

Implementing Flipped Classroom Strategy in Learning Programming

Rosnizam Eusoff, Syahanim Mohd Salleh, Abdullah Mohd Zin
Center for Software Technology and Management
Universiti Kebangsaan Malaysia
Bangi, Malaysia

Abstract—Novice students encountered many difficulties and challenges when learning to program. They face problems in terms of high cognitive load in learning and lack of prior programming knowledge. Various strategies and approaches are implemented to overcome the difficulties and challenges in programming. A flipped classroom is an active learning strategy implemented in many subjects and courses, including programming. The flipped classroom strategy consists of three phases, namely, pre-class, in-class, and post-class. A focus group discussion is conducted involving 13 participants from various learning institutions. The purpose of the study is to discuss the implementation of flipped classroom strategy in programming. The study also identifies a technique for monitoring students' involvement in activities outside the classroom and proper motivation to engage students in programming. Related research questions are constructed as guidelines for the discussion. The deductive thematic analysis is performed on the transcripts of the discussion. As a result, four pre-determine codes and two codes were generated from the analysis. This study identifies suitable activities, tools, monitoring strategies, and motivation to support the implementation of a flipped classroom in programming. There is good potential through flipped classrooms in learning programming with a systematic and careful planned implementation.

Keywords—*Flipped classroom; learning programming; cognitive load; active learning; focus group discussion*

I. INTRODUCTION

Programming is a complex subject that requires high concentration and focuses from the students. The process of learning programming requires a tremendous amount of exposure to understand the logic in each programming solution using the basic concepts despite the overwhelming syntax it might carry [1][2]. Novice students in learning the subject face difficulties and challenges. Challenges and difficulties are identified through studies in programming [3]–[7]. Among the challenges in programming are high cognitive workload [1][8], inappropriate learning strategies, time constraints, and lack of preparation before entering the classroom [2][9]. Novice students with limited knowledge need a strategy or method to help them in programming [10][11]. Active participation of the students in the learning process is one of the essential aspects of learning programming [12][13]. The difficulties experienced by students in learning can be attributed to the passive role played by them during traditional lectures [14].

Among the new strategies used in programming is flipped classroom (FC), where lecturers employ this strategy in the

learning session to enhance and improve the student learning experience [14]–[16]. The main advantages of FC are providing students with prior knowledge and preparing them before class; give extra learning time; experience active and collaborative learning in class, and strengthening the understanding of new knowledge and skills after class [17]–[19]. There are three phases of the learning process in the FC, namely pre-class, in-class, and post-class. Pre-class activities focus on theory, initial information, and preparation before entering the classroom. In-class sessions emphasize active learning activities for developing new knowledge and skills. Active learning requires students to engage in meaningful learning activities in class [20]. At the same time, post-class activities focus on strengthening knowledge and skills. Programming involves various cognitive activities and structures of multiple skills to learn [21]. FC can create a new learning environment with a practical session and enhance students' learning experience [22]. FC also enhances student satisfaction and their level of engagement in learning [23]. Besides, the implementation of FC also positively impacts students' self-efficacy and intrinsic motivation [24]. In a traditional classroom, educators allocate ample time to convey information to students to read and learn the information by themselves.

On the other hand, FC promotes pre-class activity whereby some knowledge is learned before class. This situation will give sufficient classroom time to lecturers as some of the topics are covered during the pre-class activity. Thus, lecturers can utilize the extra classroom time to focus more on impactful learning activities in programming [25]. The use of FC in programming strategy is reported to be more effective at the tertiary institution level than at the school level. Although more research on FC is increasingly conducted by researchers for various subjects [26], implementing FC in programming is not much discussed in the reports [27]. Instructors who want to use FC in their learning need to understand the implementation strategies to ensure the method's success and achieve its objectives [28]. In this regard, there is a need to conduct more research in FC [23], especially studies that describe the implementation of FC in programming.

II. METHODOLOGY

Researchers have utilized a focus group discussion (FGD) technique to obtain qualitative data, which involved discussions in small groups [29]. FGD is a suitable technique for generating new ideas compared to individual interviews

because the participants can brainstorm ideas during the ongoing discussion [30]. Besides, the FGD method is an excellent option to understand the views and opinions of others regarding the discussed subject [31]. Further, FGD is a flexible method that can be performed in a variety of conditions. There are seven steps of FGD involved in this study, as listed in Fig. 1.

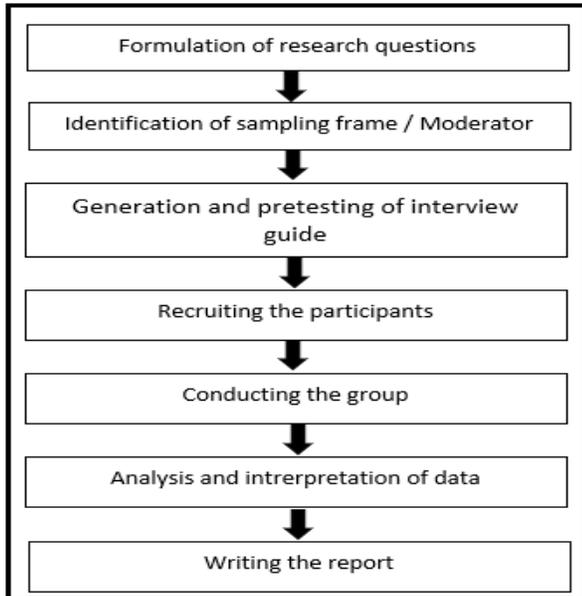


Fig. 1. Steps in Conducting a Focus Group Discussion Study. Adapted from [32], "Focus Groups Theory and Practice," p. 50.

A. Research Questions

A good research question (RQ) will guide the implementation of the study to be smoother and more efficient. Four research questions are posed in this study as follows:

RQ 1. What are the appropriate activities to implement FC strategy in programming?

RQ 2. What are the suitable tools that can support FC strategy in programming?

RQ 3. How to monitor students' participation in FC activities outside the classroom?

RQ 4. How to motivate students in learning programming?

B. Participants

The implementation of FGD requires two critical elements; the recruitment of participants and the design of the interview guide [32]. Research questions in the study are used as the design of the interview guide. Next, the researchers conducted a pilot test involving five participants. The purpose of the pilot test is to identify the weaknesses and provide improvements to the actual FGD. According to [33], the appropriate number of FGD is about six to twelve participants. The optimal time for each session is between one to two hours. Participants in the FGD are 13 senior lecturers who have been experienced in teaching programming at institutions of higher learning in Malaysia for about 12 to 32 years. According to [34], teachers with five to seven years of teaching experience can be considered expert teachers. FGD is conducted online by using

Zoom application. Participants were split into two groups to facilitate the online discussion effectively. The first group discussion was attended by six participants, and the second group was accompanied by seven participants.

C. Transcribing and Member Checking

The recording of the discussion is transcribed in a clean verbatim manner. Participants checked the transcript for the member checking process. Submission of the transcripts to the respective participants improves the accuracy of the data [35]. Furthermore, the checking process permits the participants to confirm or deny the interpretation made by the researchers on the views and suggestions of the participants during the discussion [36].

D. Coding and Analysis

Code is the smallest unit of analysis with exciting features about the data considered relevant to the research question. The data contained in the transcript is in the form of excerpts generated from the discussion. There are four predefined codes at the initial stage of the analysis process: Activity, Tools, Monitoring, and Motivation code. Two additional codes are produced when the analysis process is carried out, namely Learning Strategy and Evaluation. In total, there are 232 passages produced from the transcripts. Learning Strategy code has 91 passages which is the code that has the highest passages.

Meanwhile, the Activity code consists of 53 passages, and the Tools code consists of 36 passages. The Motivation code consists of 31 passages, while the Monitoring consists of 15 passages. Lastly, the Evaluation code consists of 6 passages. Thematic analysts are used to analyzing these coded passages. Thematic analysis analyses and translates qualitative data to form a specific theme [37].

III. FINDINGS

A. Activities to Implement FC Strategy in Programming

In general, the findings in this section are made through passages grouped under the Learning Strategies, Activity and Motivation codes. As mentioned, FC comprises three phases, namely pre-class, in-class, and post-class. The learning activities involved can be regarded as the same or slightly different for each of these phases. However, students may encounter issues when learning to program in FC strategy. The problems are caused by students' failure to understand FC requirements. Hence, during the first face-to-face class meeting with the students, a sufficient explanation regarding the implementation of FC in learning should be given to the students. Students are informed about the three essential phases involved in FC: pre-class, in-class, and post-class. In the early stages of the course, students must participate in the learning session and eliminate their anxiety about programming. Lecturers can share exciting videos about programming with students in the class as a first insight for them. "The first topic is crucial to attracting students to programming. If students think that the first topic is difficult, most probably, students will have problems engaging with other topics. Students should not be addressed with difficult questions, but instead, the lecturer may opt for intermediate questions when asking questions to the respective students."

Ensuring students' participation in learning activities outside the classroom is the biggest challenge the educators face when implementing FC strategy [38]–[42]. A participant suggested that allocation of marks should be given to students who are actively involved in the pre-class and post-class activities. According to the participant, extra marks will motivate students to engage in activities outside the classroom actively. Thus, lecturers should consider allocating marks for students who show good involvement during the pre-class and post-class activities. This initiative will attract students' interest and boost their motivation in the learning process. "From my experience all this while, students will only do extra assignments outside class if there are only marks given for those assignments. I believe some extra marks will motivate them well."

Although students may view videos about programming online, lecturers need to guide their students on which video they should watch. A meaningful learning environment should be created as much as possible in learning activities. For example, the teacher should design the questions and problems based on students' daily life experiences, such as using ATMs and vending machines or developing a smartphone application's software. Most students have anxiety based on the information they have obtained about the difficulties of programming. "Lecturers need to create a meaningful learning environment for students. These novice students may have zero knowledge of programming. Most of them are worried about programming."

Programming is known as one of the subjects that has a high percentage of failures among students. The learning activities in programming should gradually develop students' skills from easy to challenging levels. The aim for each activity will focus only on one new skill so that students will not experience a high cognitive load. In the early stages of learning, students are exposed to videos from various sources to inform them about the importance of programming and its application in daily life. The video viewing activities can also be assigned to students as pre-class activities. Next, as in the class session, students are divided into several groups, and they are asked to discuss the videos they have watched and present the results of their discussions in class.

Lecturers can prepare several sets of questions for the students to discuss in their groups. Every two groups will receive a similar question for discussion. Then, each group should present the results of the meeting to all. Through this method, two different solutions can be generated from the same question. Based on this, students will be able to perceive that a problem that occurs in programming has various solutions. Post-class activities focus on exercises and tasks to strengthen students' new knowledge and skills. As a guide, the lecturer may provide students with examples of complete solutions to the assigned questions. Next, students will receive questions that are similar to the given example as post-class activity. Students can refer to the provided examples as a guide to answer the questions. These activities can be assigned as individual or group assignments. Thus, students will not feel burdened to carry out post-class activities through this method, especially for challenging problems.

B. Suitable Tools to Support FC in Programming

The findings in the section are based on passages grouped under the Tools and Motivation codes. In supporting a learning process, lecturers can incorporate myriad choices of tools and supporting materials. Hence, to implement the FC strategy in the learning session, lecturers must prepare and choose suitable tools and learning materials for the class, especially for the pre-class and post-class activities, as the lecturer's supervision is absent. Video is the most widely used material in FC at the pre-class level [39]. The videos can be self-produced by lecturers or taken from various sources on the internet, such as YouTube. The optimum period for video viewing is between five to ten minutes [39] [43]. An appropriate video screening period will ensure that students are not burdened with such activity. Lecturers may assign quizzes or short questions to students while watching the video or after finishing the video. The task will ensure that students will pay attention to the details in the video as they need to answer the quiz or short questions. Besides that, lecturers can provide the students with a worksheet to record the pre-class and post-class activities. These worksheets can also be used to assess students' involvement in activities and class discussions. The worksheets are applicable in a lab, so students can use them to mark errors they have encountered while running a program.

Apart from providing students with questions, lecturers can ask students to look and solve the problems by themselves. The problem-solving tasks can be regarded as project work. For example, students are asked to identify the business needs of specific companies through the company's advertising catalogue. Students must identify the problem and suggest a solution using programming. In the early stages, lecturers can help students by providing ideas and input. Once students already have solid knowledge, they are encouraged to figure out their solutions without the intervention of the lecturer. The use of infographic notes should be given priority on technical topics in programming. The infographic form notes are easier to understand than text notes, especially in specialized topics such as control structures, arrays and method. "After the introduction topic, I think infographic notes are more suitable to be used because it will be easier for students to see the whole process in programming through infographics as compared to usual text-based notes."

C. Monitoring Students' Participation in FC

The discussion of the findings in this section is based on the passages grouped under the Monitoring and Evaluation codes. Class management is a critical aspect of a physical classroom, while learning materials are essential online [44]. The biggest challenge in FC is to ensure that student engagement in activities outside the classroom [38][45]. The outside classroom's activities will occur without supervision from lecturers. In this situation, lecturers can measure students' involvement in the pre-class activities by judging students' responses in the class. Usually, students who have prepared before class will be more ready and able to answer questions while in-class compared to students who don't prepare. At the beginning of class, lecturers can ask some questions that are related to the pre-class activity. Students who have not completed their pre-class activities will be unable to answer the question. Monitoring students' involvement in activities is also

more accessible through a LMS platform such as Moodle. Log-in records into the LMS can be regarded as proof of students' participation outside of the classroom. A short quiz can be embedded in the video to ensure students will watch the entire video without skipping. In this way, students are required to watch the whole video to answer the quiz.

Besides, monitoring also can be done by students themselves in group activities. Every group has an appointed leader, and the leader is responsible for reminding the group members about the assigned task. Most of the students are more receptive to reprimands from their peers compared to lecturers. "From my experience, students have more of a sense of guilt towards their friends than lecturers. Most of their daily life is with their friends in the hostel. They would be embarrassed if they did not participate in the activities with their group."

Monitoring students' involvement in pre-class and post-class activities is vital in the implementation of the FC strategy. However, participants' opinion that students should be allowed to do activities outside the classroom at their convenience. In this regard, lecturers need to determine the appropriate monitoring method to be applied to the students accordingly. The most important thing is to make sure students carry out the activities they should do and complete outside the classroom.

D. Motivating Students in Learning Programming

The discussion of this section's findings is based on the passages grouped under the Motivation and Evaluation codes. Motivation is a crucial element in determining success in learning. Successful students are usually students who are highly motivated in their learning. Researchers have conducted various studies that show a connection between motivation and success in learning programming [5][46][47]. The different approaches and strategies introduced in the learning increase student motivation and make programming more exciting and relevant to real-life situations [48]. There are two types of motivation: intrinsic motivation from internal factors and extrinsic motivation from external factors [49]. Most of the motivational methods that are given to students are extrinsic forms of motivation. However, the best motivation is the intrinsic type of motivation, which comes from the students themselves. However, intrinsic motivation can also be developed from extrinsic motivation. For example, when students always achieve high marks in tests or quizzes, their interest in programming will subsequently increase, and this is where intrinsic motivation will indirectly appear. A study by [41], found that at the early stages of the implementation of FC in learning, students are more motivated to do assignments or quizzes as activities outside the classroom. These two activities are considered as extrinsic motivation for students. Gradually, intrinsic motivation will be formed when students are motivated to engage with any FC's activities outside the classroom. The success of new approaches in learning will occur if students are instilled with more intrinsic motivation. Lecturers also need to constantly motivate students about the importance of programming in student's daily lives. However, based on experience, most students will carry out activities directed by their extrinsic motivation. For example, students will only complete an assignment when they are informed that the marks will be awarded for the work. In this regard, it is

advisable if the allocation of marks is given for outside classroom activities. "My experience with students so far, if they knew there would be no marks for an activity, they would not do it."

As the most crucial factor in determining the success of flipped classroom strategies introduced to students, the motivational element needs attention in learning. Lecturers need to be creative in finding suitable methods to increase the motivation of their students.

IV. DISCUSSION

In general, FC is a technology-based strategy in providing early exposure to students outside the classroom. Students, in turn, will gain in-depth exposure during the classroom session [50][51]. Comparison of four learning strategies used in [52] study has found that FC strategy produced students with best performance achievement compared to other strategies. There are various methods of FC strategies that can be implemented in the learning process. The main goal of the FC strategy is to provide students with ample learning time and the opportunity to gain early exposure to the learning content before the class session [53]. The use of FC strategies does not mean eliminating the role of the instructor or any form of direct instruction that occurs in a learning process. Instead, this strategy maximizes meeting time between instructor and student with strategic activities that have a higher impact on learning programming. FC can also meet the individual needs of different students without having to reduce the material and filling in the learning [54]. According to [51], students who received FC exposure are more sensitive to their learning processes than students exposed to the traditional strategy. Several essential elements need to be presented in the FC implementation model for programming. The main elements are strategic planning of activities according to the FC phases, preparation of appropriate tools for learning activities, and an effective monitoring strategy for outside classroom activities. Student motivation is a key element in determining the success of the FC strategy used. Therefore, lecturers need to provide motivation to attract students to programming. Intrinsic motivation is best, although it is quite difficult to establish in the student. However, extrinsic motivation can turn into intrinsic motivation gradually as students begin to take an interest in programming. A flipped classroom model in programming is illustrated in Fig. 2.

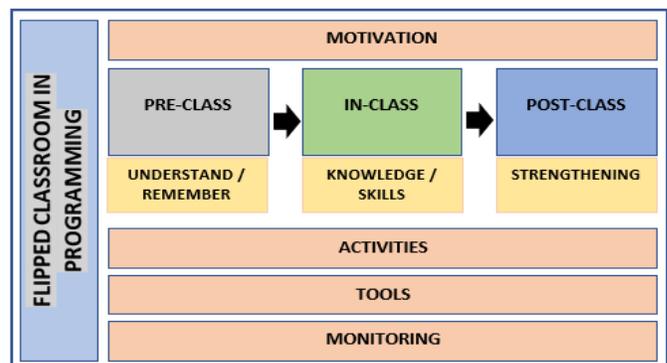


Fig. 2. A Flipped Classroom Model in Programming.

V. CONCLUSION

There are myriad advantages gained through the implementation of FC in learning programming. Among the benefits are that students will be well prepared before entering the classroom as they must conduct a pre-class activity. This condition will aid the learning process in the classroom to be more efficient because students are equipped with basic knowledge. An extra-time earned can be used to conduct strategic active and collaborative learning in the classroom. The main goal of FC is to create a student-centred active learning environment. The lecturer only serves as a moderator who monitors the learning activities in the classroom. Post-class activities will focus on strengthening new knowledge that has been learned in class. However, the implementation of this FC strategy needs to be planned carefully and strategically to get the full advantage of this strategy.

ACKNOWLEDGMENT

The authors acknowledge Universiti Kebangsaan Malaysia and the Ministry of Education Malaysia for their support. This study was funded by the UKM Research Grant (GGPM-2020-027).

REFERENCES

- [1] S. M. Salleh, Z. Shukur, and H. M. Judi, "Scaffolding Model for Efficient Programming Learning Based on Cognitive Load Theory," *Int. J. Pure Appl. Math.*, vol. 118, no. 7 Special, pp. 77–82, 2018.
- [2] K. M. Yusoff, N. S. Ashaari, T. S. M. T. Wook, and N. M. Ali, "Validation of the Components and Elements of Computational Thinking for Teaching and Learning Programming using the Fuzzy Delphi Method," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 80–88, 2021.
- [3] S. M. Shuhidan, M. Hamilton, and D. D'Souza, "Understanding novice programmer difficulties via guided learning," *Proc. 16th Annu. Jt. Conf. Innov. Technol. Comput. Sci. Educ. - ITiCSE '11*, p. 213, 2011.
- [4] B. Özmen and A. Altun, "Undergraduate Students' Experiences in Programming: Difficulties and Obstacles," *Turkish Online J. Qual. Inq.*, vol. 5, no. 3, pp. 9–27, 2014.
- [5] A. Gomes and A. Mendes, "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations," *Proc. - Front. Educ. Conf. FIE*, vol. 2015-Febru, no. February, 2015.
- [6] Y. Qian and J. Lehman, "Students' Misconceptions and Other Difficulties in Introductory Programming," *ACM Trans. Comput. Educ.*, vol. 18, no. 1, pp. 1–24, 2017.
- [7] M. Rahmat, S. Shahrani, R. Latih, N. F. M. Yatim, N. F. A. Zainal, and R. A. Rahman, "Major Problems in Basic Programming that Influence Student Performance," *Procedia - Soc. Behav. Sci.*, vol. 59, pp. 287–296, 2012.
- [8] R. Duran, J. Sorva, and S. Leite, "Towards an analysis of program complexity from a cognitive perspective," *Proc. 2018 ACM Conf. Int. Comput. Educ. Res. - ICER '18*, pp. 21–30, 2018.
- [9] E. Lahtinen, K. Ala-mutka, and H.-M. Jarvinen, "A Study of the Difficulties of Novice Programmers," *ACM ITiCSE'05*, pp. 14–18, 2005.
- [10] F. Layth Khaleel, N. Sahari Ashaari, and T. S. M. Tengku Wook, "An Empirical Study on Gamification for Learning Programming Language Website," *J. Teknol.*, vol. 81, no. 2, 2019.
- [11] A. R. Mohamad Gobil, Z. Shukur, and I. A. Mohtar, "Novice difficulties in selection structure," *Proc. 2009 Int. Conf. Electr. Eng. Informatics, ICEEI 2009*, vol. 2, no. August, pp. 351–356, 2009.
- [12] S. B. Ho, S. L. Chean, I. Chai, and C. H. Tan, "An Assessment of Learning Content Model for Introductory Programming in Higher Education," *ASM Sci. J.*, vol. 14, no. 1, pp. 55–62, 2021.
- [13] C. Stöhr and T. Adawi, "Flipped Classroom Research: From 'Black Box' to 'White Box' Evaluation," *Educ. Sci.*, vol. 8, no. 1, p. 22, 2018.
- [14] M. N. Giannakos, J. Krogstie, and D. Sampson, "Putting Flipped Classroom into Practice: A Comprehensive Review of Empirical Research," *Digit. Technol. Sustain. Innov. Improv. Teach. Learn.*, pp. 27–44, 2018.
- [15] S. R. Sobral, "Flipped classrooms for introductory computer programming courses," *Int. J. Inf. Educ. Technol.*, vol. 11, no. 4, pp. 178–183, 2021.
- [16] K. Thongkoo, P. Panjaburee, and K. Daungcharone, "Integrating inquiry learning and knowledge management into a flipped classroom to improve students' web programming performance in higher education," *Knowl. Manag. E-Learning*, vol. 11, no. 3, pp. 304–324, 2019.
- [17] J. L. Jensen, E. A. Holt, J. B. Sowards, T. Heath Ogden, and R. E. West, "Investigating Strategies for Pre-Class Content Learning in a Flipped Classroom," *J. Sci. Educ. Technol.*, vol. 27, no. 6, pp. 523–535, 2018.
- [18] W. Kelly, "Flipping the Classroom to Solve the Time Problem," 2017. [Online]. Available: https://flippedlearning.org/flexible_environment/flipping-classroom-solve-time-problem/. [Accessed: 20-Nov-2018].
- [19] J. Herala, A. Vanhala, E. Knutas, A., & Ikonen, "Teaching programming with flipped classroom method: a study from two programming courses," *15th Koli Call. Conf. Comput. Educ. Res.*, pp. 165–166, 2015.
- [20] B. Sohrabi and H. Iraj, "Implementing flipped classroom using digital media: A comparison of two demographically different groups perceptions," *Comput. Human Behav.*, vol. 60, pp. 514–524, 2016.
- [21] R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Trans. Educ.*, pp. 1–14, 2018.
- [22] K. Siripongdee, P. Pimdee, and S. Tuntiwongwanich, "A blended learning model with IoT-based technology: Effectively used when the COVID-19 pandemic?," *J. Educ. Gift. Young Sci.*, vol. 8, no. 2, pp. 905–917, 2020.
- [23] G. Akçayır and M. Akçayır, "The flipped classroom: A review of its advantages and challenges," *Comput. Educ.*, vol. 126, no. January, pp. 334–345, 2018.
- [24] T. N. T. Thai, B. De Wever, and M. Valcke, "The impact of a flipped classroom design on learning performance in higher education: Looking for the best 'blend' of lectures and guiding questions with feedback," *Comput. Educ.*, 2017.
- [25] J. Elmaleh, "Improving Student Learning in an Introductory Programming Course Using Flipped Classroom and Competency Framework," in *IEEE Global Engineering Education Conference, EDUCON, 2017*, no. April, pp. 49–55.
- [26] C. K. Lo and K. F. Hew, "A critical review of flipped classroom challenges in K-12 education: possible solutions and recommendations for future research," *Res. Pract. Technol. Enhanc. Learn.*, vol. 12, no. 1, p. 4, 2017.
- [27] Hendrik and A. Hamzah, "Flipped Classroom In Programming Course: A Systematic Literature Review," *Int. J. Emerg. Technol. Learn.*, vol. 16, no. 2, pp. 220–236, 2020.
- [28] Z. Turan and B. Akdag-cimen, "Flipped classroom in English language teaching: a systematic review Flipped classroom in English language teaching: a systematic review," *Comput. Assist. Lang. Learn.*, vol. 0, no. 0, pp. 1–17, 2019.
- [29] A. J. Onwuegbuzie, W. B. Dickinson, N. L. Leech, and A. G. Zoran, "A Qualitative Framework for Collecting and Analyzing Data in Focus Group Research," *Int. Inst. Qual. Methodol.*, pp. 1–21, 2009.
- [30] R. L. Breen, "A practical guide to focus-group research," *J. Geogr. High. Educ.*, vol. 30, no. 3, pp. 463–475, 2006.
- [31] S. Wilkinson, "Focus group methodology: a review," *Int. J. Soc. Res. Methodol.*, vol. 1, no. 3, pp. 181–203, 1998.
- [32] D. W. Stewart and P. N. Shamdasani, *Focus Groups Theory and Practice*, Third. SAGE Publications Inc, 2014.
- [33] A. Nili, M. Tate, and D. Johnstone, "A framework and approach for analysis of focus group data in information systems research," *Commun. Assoc. Inf. Syst.*, vol. 40, no. December, pp. 1–21, 2017.

- [34] D. C. Berliner, "Expert Teachers : Their Characteristics , Development and Accomplishments," pp. 13–28, 2001.
- [35] L. Birt, S. Scott, D. Cavers, C. Campbell, and F. Walter, "Member Checking: A Tool to Enhance Trustworthiness or Merely a Nod to Validation?," *Qual. Health Res.*, vol. 26, no. 13, pp. 1802–1811, 2016.
- [36] A. G. Candela, "Exploring the function of member checking," *Qual. Rep.*, vol. 24, no. 3, pp. 619–628, 2019.
- [37] V. Clarke and V. Braun, "Thematic analysis," *J. Posit. Psychol.*, vol. 12, no. 3, pp. 297–298, 2017.
- [38] A. Mohamed, "Designing a CS1 Programming Course for a Mixed-Ability Class," in *WCCCE '19*, 2019, pp. 10–15.
- [39] T. Ishak, R. Kurniawan, Z. Zainuddin, and C. M. Keumala, "The role of pre-class asynchronous online video lectures in flipped-class instruction : identifying students ' perceived need satisfaction," *J. Pedagog. Res.*, pp. 1–11, 2019.
- [40] Y.-H. Chang, A.-C. Song, and R.-J. Fang, "Integrating ARCS Model of Motivation and PBL in Flipped Classroom: a Case Study on a Programming Language," *EURASIA J. Math. Sci. Technol. Educ.*, vol. 14, no. 12, 2018.
- [41] V. Gupta, "Blended SPOC Teaching and Learning Model for Computer Programming Course: Insights and Defeating Challenges," *IEEE TALE2020 - An Int. Conf. Eng. Technol. Educ.*, pp. 251–257, 2020.
- [42] H. Y. Durak, "Modeling Different Variables in Learning Basic Concepts of Programming in Flipped Classrooms," *J. Educ. Comput. Res.*, 2019.
- [43] K. Zeuch, S. Kaven, and V. Skwarek, "Evaluation of a re-designed introductory course " Programming in C " with video support," in 2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET), 2019, pp. 1–6.
- [44] F. A. Albrahim, "Online Teaching Skills and Competencies," *TOJET Turkish Online J. Educ. Technol.*, vol. 19, no. 1, pp. 9–20, 2020.
- [45] H. Y. Durak, "Flipped learning readiness in teaching programming in middle schools : Modelling its relation to various variables," *J. Comput. Assist. Learn.*, no. July, pp. 939–959, 2018.
- [46] S. Alhazbi, "Using flipped classroom approach to teach computer programming," *Proc. 2016 IEEE Int. Conf. Teaching, Assess. Learn. Eng. TALE 2016*, no. December, pp. 441–444, 2016.
- [47] S. Nikolic, M. Ros, and D. B. Hastie, "Teaching programming in common first year engineering: discipline insights applying a flipped learning problem-solving approach," *Australas. J. Eng. Educ.*, pp. 1–12, 2018.
- [48] R. A. Alturki, "Measuring and improving student performance in an introductory programming course," *Informatics Educ.*, vol. 15, no. 2, pp. 183–204, 2016.
- [49] E. L. Deci and R. M. Ryan, "Self-determination theory: A macrotheory of human motivation, development, and health," *Can. Psychol.*, vol. 49, no. 3, pp. 182–185, 2008.
- [50] J. L. Bishop and M. A. Verleger, "The Flipped Classroom A Survey Of The Research," in 120th ASEE Annual Conference & Exposition, 2013.
- [51] J. F. Strayer, "How learning in an inverted classroom influences cooperation, innovation and task orientation," *Learn. Environ. Res.*, vol. 15, no. 2, pp. 171–193, 2012.
- [52] N. T. T. Thai, B. De Wever, and M. Valcke, "The impact of a flipped classroom design on learning performance in higher education: Looking for the best 'blend' of lectures and guiding questions with feedback," *Comput. Educ.*, vol. 107, pp. 113–126, 2017.
- [53] C. E. Davenport, "Evolution in Student Perceptions of a Flipped Classroom in a Computer Programming Course," *J. Coll. Sci. Teach.*, vol. 47, no. 4, pp. 30–35, 2018.
- [54] N. Hamdan, P. McKnight, K. McKnight, and K. M. Arfstrom, "A Review of Flipped Learning," vol. 15, no. 5, pp. 86–87, 2013.